Problem 1a:

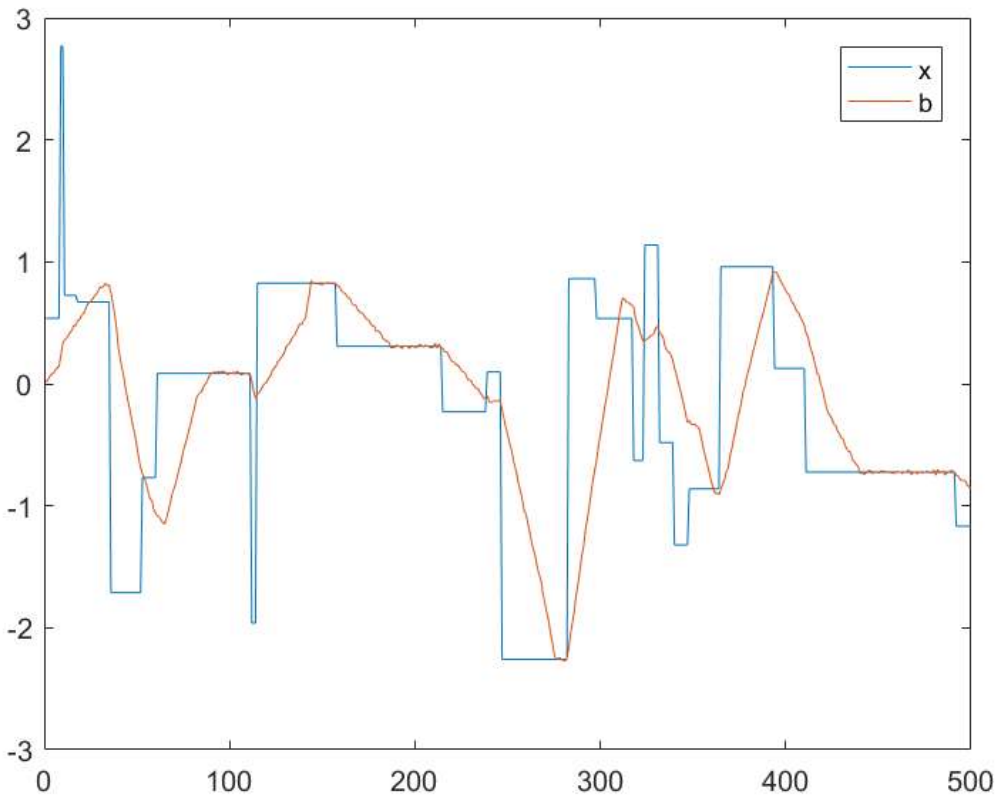
```
function A = genA(n,k)
A = zeros(n);
for i = 1:n
    if i-k<1
        a = 1;
    else
        a = i-k+1;
    end
    for j = a:i
        A(i,j) = 1/k;
    end
end
end

```

Assignment 7

Contents

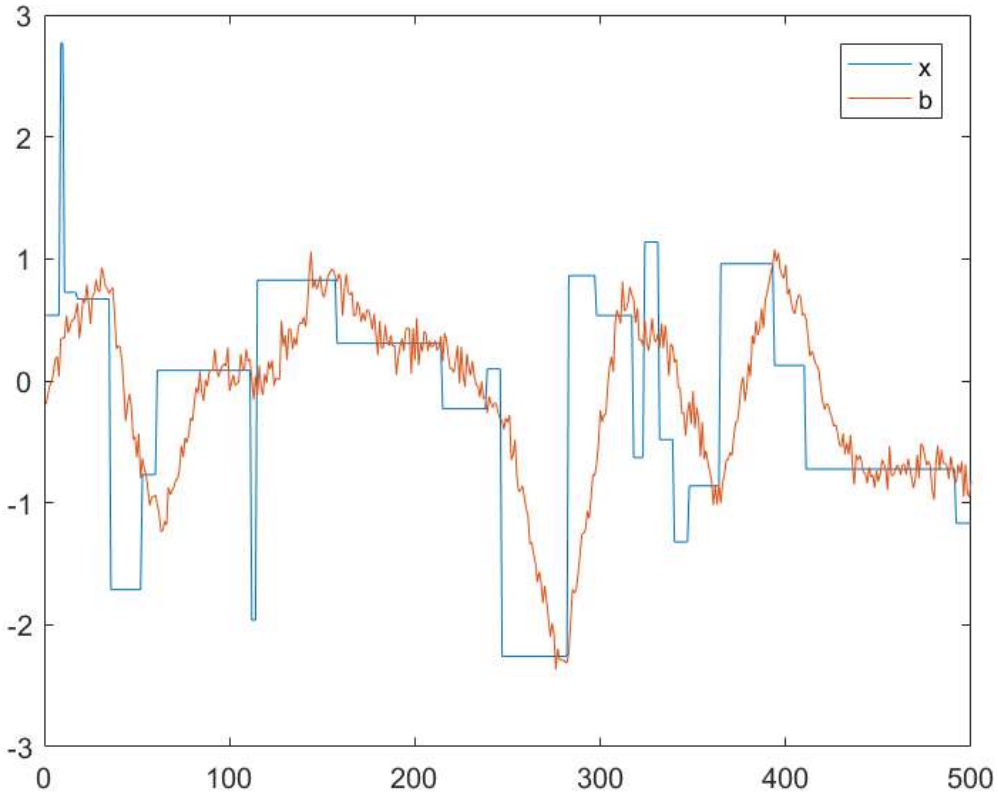
Problem 1b sigma = 0.01

```
x = csvread('xsignal.csv');
A = genA(500,30); % since x is 500*1
w = 0.01*randn(500,1);
b = A*x+w;

plot(x);
hold on;
plot(b);
legend('x','b');
hold off
```



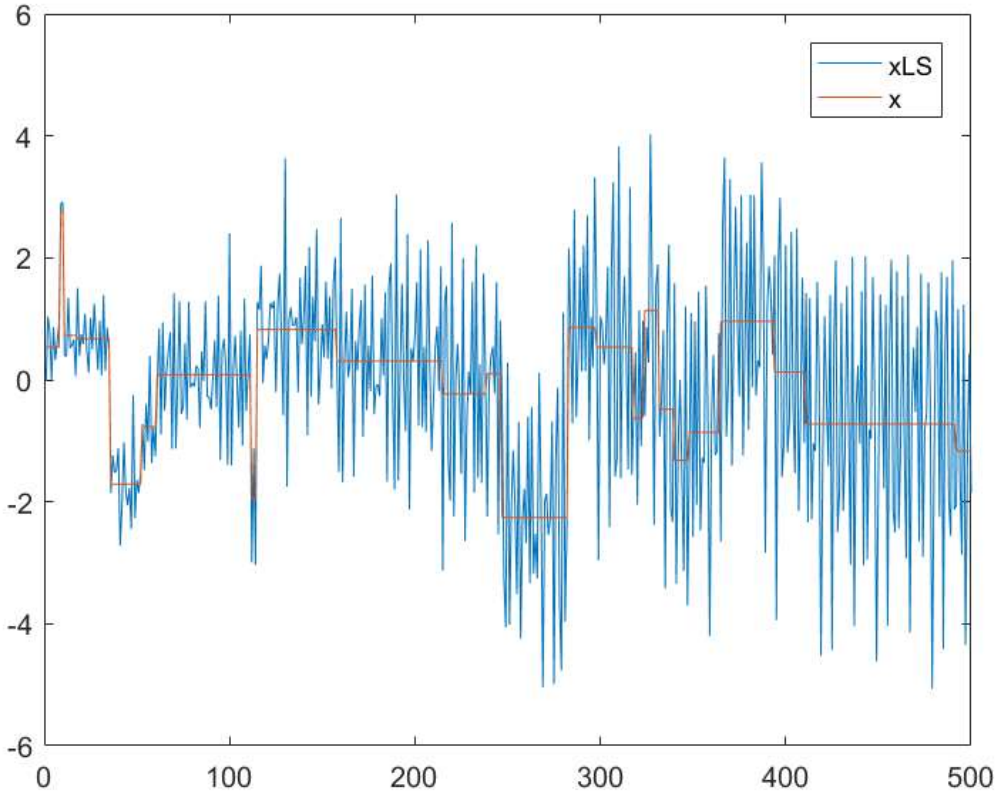
Problem 1b sigma = 0.1

```
x = csvread('xsignal.csv');
A = genA(500,30); % since x is 500*1
w = 0.1*randn(500,1);
b = A*x+w;

plot(x);
hold on;
plot(b);
legend('x','b');
hold off
```



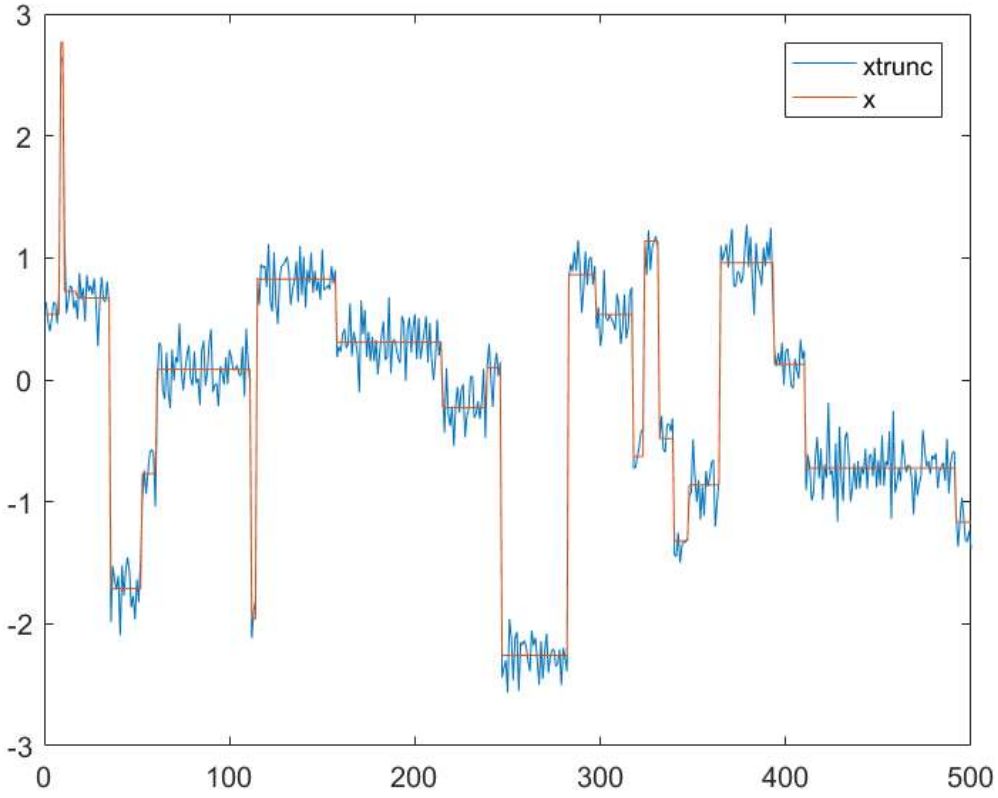
Problem 1c Ordinary Least Squares

```
x = csvread('xsignal.csv');
A = genA(500,30);
w = 0.01*randn(500,1); % here let sigma=0.01
xLS = x + ((transpose(A)*A)^(-1))*transpose(A)*w;

plot(xLS);
hold on;
plot(x);
legend('xLS','x');
hold off;
```



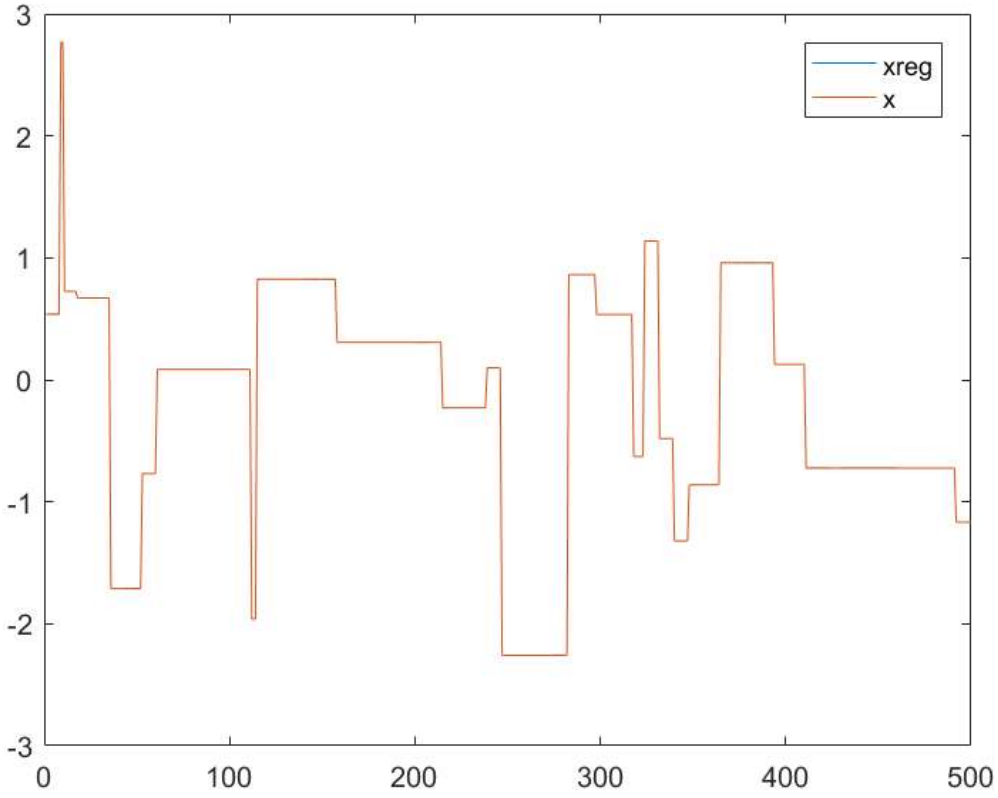
Problem 1c Truncated SVD

```
x = csvread('xsignal.csv');
A = genA(500,30);
[U,S,V] = svd(A);
m = 300;
S_new = zeros(length(S));
for i=1:m
    S_new(i,i) = 1.0/S(i,i);
end

w = 0.01*randn(500,1); % here let sigma=0.01
x_trunc = x + V*S_new*transpose(U)*w;
plot(x_trunc);
hold on;
plot(x);
legend('xtrunc','x');
hold off
```



Problem 1c Regularized

```
x = csvread('xsignal.csv');
A = genA(500,30);
[U,S,V] = svd(A);
m = 300;
lambda = 3;
S_new = zeros(length(S));
for i=1:m
    S_new(i,i) = (1.0*S(i,i))/((S(i,i))*(S(i,i))+lambda);
end

w = 0.01*randn(500,1); % here let sigma=0.01
x_reg = x + V*S_new*transpose(U)*w;
plot(x_reg);
hold on;
plot(x);
legend('xreg','x');
hold off
```



Problem 1c interpretation

Based on three different results generated by three different methods in part c, regularized least squares is the best one, then truncated SVD is the second best, and ordinary least squares is the third. With larger m, the prediction will be less accurate. When lambda is larger, it is more accurate.

Problem 1d

When experimenting different values of k, I found that with smaller k the accuracy increases. Lower noise level results in a more accurate method.

Published with MATLAB® R2016b